Forschungsbericht 2016-04

Standardisierung eines erweiterbaren Modells für Provenance-Daten (PROV-SPEC)

Andreas Schreiber

Deutsches Zentrum für Luft- und Raumfahrt
Simulations- und Softwaretechnik
Köln-Porz, Berlin-Zentrum

DLR Deutsches Zentrum für Luft- und Raumfahrt

Inhalt

Projekt PROV-SPEC

Die Aufzeichnung detaillierter Informationen über die Entstehung und den Ursprung von Daten wird als Provenance bezeichnet [1]. Basierend auf einem Modell des aufzuzeichnenden Prozesses („Provenance-Modell“) werden dabei alle relevanten Informationen des Prozesses in einer Datenbank gespeichert. Diese können dann analysiert und entsprechend der interessierenden Fragestellen ausgewertet werden.

Ziel des Vorhabens ist die Erstellung eines Vorschlags für einen nationalen Standard eines offenen Provenance-Modells. Dieser Vorschlag soll auf Basis bereits bestehender offener Provenance-Modelle erstellt werden. Insbesondere auf Grundlage des Open Provenance Modells (OPM) [2] und dessen Weiterentwicklung Provenance Data Model (PROV-DM) [3]. Im Projekt sollen dazu diese existierenden Modelle evaluiert und ggf. an nationale Bedürfnisse angepasst werden. Im Gegenzug sollen Vorschläge und Ideen in die internationalen Standardisierungsgremien, insbesondere das World Wide Web Consortium (W3C), eingebracht werden.

Der Vorschlag für einen nationalen Standard soll praxisrelevant, praktikabel und einsatzbereit sein. Die Praxisrelevanz soll durch Evaluation mit praxis- und industrienahen Anwendungen verschiedener Branchen bewiesen werden (im Vorhaben insbesondere durch Anwendungen aus den Ingenieurwissenschaften, der Medizin und der Klima- und Meeresforschung). Praktikabilität soll durch möglichst einfach und anwendungsnah gestaltete Methodiken und Schnittstellen zu Provenance-Datenbanken erreicht werden. Die Einsatzbereitschaft soll durch Bereitstellung einer Provenance-Datenbank als frei verfügbare Open-Source-Software und durch Bereitstellung eines übersichtlichen Handbuchs hergestellt werden.

Zusammengefasst waren die Ziele des Projekts:

- Erstellung eines Vorschlags für ein DIN SPEC für ein offenes Provenance-Modell.
- Erstellung und Dokumentation einer Methodik zur Einführung von Provenance-Aufzeichnung in existierende und geplante Prozesse.
- Bereitstellung einer frei verfügbaren Provenance-Datenbank als Referenzimplementierung.

Voraussetzungen, unter denen das Vorhaben durchgeführt wurde

Ein standardisiertes Modell für Provenance-Daten ergibt ein hohes Maß an Interoperabilität zwischen Firmen und Institutionen. Die aufgezeichneten Provenance-Daten bei Kooperationspartnern, Zulieferern oder Unterauftragnehmern können so zu den Process Ownern (z.B. das auftraggebende Unternehmen) mitgeliefert werden. Diese können dann bei der Analyse des Prozesses mit einbezogen werden, z.B. wenn Fehler auftreten oder Prozessaudits durchgeführt werden sollen. Durch die Standardisierung ergeben sich in solchen Fällen Einsparungen in den Unternehmen, da eine Konvertierung der Provenance-Daten dann nicht notwendig ist.

Planung und Ablauf des Vorhabens

Im Vorhaben sollte ein Vorschlag für die Standardisierung eines erweiterbaren Modells für Provenance-Informationen erarbeitet werden. Die Anwendungsdomäne sollte ursprünglich die Standardisierung des Modells für ein elektronisches Laborbuch sein [4, 5], das der Richtlinie eines guten Laborbuches (des OECD) folgt. Anhand dieses exemplarischen Anwendungsfalls sollte die Tauglichkeit des zu standardisierenden Provenance-Modells evaluiert werden. Dies wurde im Verlauf des Vorhabens geändert; stattdessen wurde das Provenance-Modell PROV-DM anhand von Softwareentwicklungsprozessen und

anhand eines Teilbereichs von Gesundheitsdaten evaluiert.

Neben einem Standardisierungsansatz sollten in diesem Vorhaben auch geeignete Werkzeuge, die bereits im DLR im Einsatz sind, entsprechend weiterentwickelt werden. Dazu gehört u.a. die Erstellung einer Referenzimplementierung für eine Provenance-Datenbank, die als Open-Source-Software allen Anwendern zur Verfügung gestellt werden soll.

Stand der Wissenschaft und Technik vor Vorhabensbeginn

Provenance (dt. *Provenienz*) beschreibt die detaillierte, nachvollziehbare Historie von Dingen. Im wissenschaftlichen und technischen Umfeld ist mit Provenance gemeint, das Ergebnisse so ausreichend dokumentiert sind, dass sie nachvollziehbar und reproduzierbar sind. Dies kann man beispielsweise durch detailliertes Aufzeichnen aller relevanten Arbeitsschritte in wissenschaftlich-technischen Prozessen erreichen. Idealerweise geschieht diese Aufzeichnung automatisch und ist im Falle von Prozessen, die durch Softwaresysteme realisiert sind, fest in die Software integriert.

Die Aufzeichnung aller Schritte eines Prozesses erfolgt in der Regel durch Speicherung aller wichtigen und notwendigen Informationen in einer dafür entwickelten Datenstruktur. Diese Datenstruktur nennt man *Provenance-Modell*. Das Provenance-Modell beschreibt die Repräsentation der Provenance-Daten eines Prozesses im Computer.

Vor Beginn des Vorhabens fehlten standardisierte Provenance-Modelle. Auch eine standardisierte Methodik zum Einführen von Provenance-Aufzeichnung bestehender Prozesse gab es nicht. Insbesondere war es nicht möglich, Informationen zwischen zwei verschiedenen Provenance-Datenbanken auszutauschen. Es gab zwar schon länger eine große Anzahl unterschiedlicher

Provenance-Modelle, aber die Gemeinsamkeiten sind oftmals gering, angefangen von den grundlegenden Begriffen bis hin zur genauen Semantik der Informationen. Es fehlte also ein Ansatz oder Standard um Interoperabilität zwischen den Provenance-Datenbanken verschiedener Prozesse herzustellen. Ebenso wurden und werden unterschiedlichste Technologien für die softwaretechnische Implementierung von Provenance-Datenbanken verwendet. Zum Beispiel werden relationale Datenbanken, XML-Datenbanken oder Graph-basierte Datenbanken eingesetzt. Außerdem kommen unterschiedlichste Schnittstellen zum Hinzufügen von Provenance-Informationen in die Datenbank und zur Abfrage der Provenance-Informationen zum Einsatz.

Um die verschiedenen Ansätze zur Repräsentation von Provenance-Daten besser zu verstehen, ihre Gemeinsamkeiten zu erkennen und Gründe für Unterschiede herauszufinden wurde im Jahr 2006 die sog. „Provenance Challenge" durchgeführt [6]. Als ein Ergebnis entstand eine Spezifikation für ein erstes vereinheitlichtes Provenance-Modell, das *Open Provenance Model* (OPM). In weiteren Schritten der Provenance Challenge wurde das OPM evaluiert (z.B. in Bezug auf Interoperabilität) und weiter entwickelt. Im Jahr 2010 führten diese Aktivitäten zur Etablierung der W3C Provenance Working Group, in dessen Rahmen das Provenance Data Model (PROV-DM) als Nachfolger des OPM entwickelt wird.

Eigene Arbeiten mit dem Open Provenance Model

Seit 2006 wurden in der Abteilung Verteilte Systeme und Komponentensoftware eine Reihe von Arbeiten auf Basis des OPM durchgeführt. Ziel war, Erfahrungen mit dem OPM zu sammeln und sein Tauglichkeit für den praktischen Einsatz zu evaluieren. Einige Arbeiten hatten daneben reinen Forschungscharakter.

Die durchgeführten Arbeiten waren:

- Aufzeichnen von Provenance-Informationen aus Python-Skripten durch

aspektorientierte Programmierung [7]
- Analysieren vom grundlegenden Anforderungen für das Analysieren von Provenance [8, 9]
- Erstellen eines Provenance-Modells für verteilte Simulations-Workflows und Integrieren von Provenenance-Aufzeichnung in die Integrations- und Workflowumgebung TENT [10]
- Entwickeln eines elektronischen Laborbuchs zur nachvollziehbaren Dokumentation von Experimenten und Simulationen [11, 12]
- Erstellen eines Provenance-Modells für Softwareentwicklungsprozesse und exemplarisches Aufzeichnen von Provenance-Informationen von Teilprozessen [13-15]
- Ermitteln grundlegender Anforderungen an Komponenten zur Visualisierung von Provenance-Informationen [8, 9]
- Entwickeln einer Client-Bibliothek zum Aufzeichnen von Provenance-Informationen in Python [16]
- Entwickeln eines Services zum Speichern von Provenance-Informationen in der Graph-Datenbank Neo4j [17]

Zusammenarbeit mit anderen Stellen

Die Arbeiten wurden in Abstimmung mit der W3C Provenance Working Group durchgeführt. Es ließ sich nicht realisieren, dass das DLR Mitglied im W3C wird. Dies wäre notwendig gewesen, um formell in den Gremien des W3C mitarbeiten zu können. Daher war vom DLR niemand bei den formellen Treffen der Provenance Working Group dabei.

Unabhängig davon, hat das DLR mit wesentlichen Experten auf der Gebiet der Provenance-Forschung zusammen gearbeitet. Insbesondere mit der University of Southampton (Prof. Luc Moreau und Mitarbeiter) und die VU Amsterdam (Dr. Paul Groth; inzwischen bei Elsevier), zu denen bereits langjährige Kontakte bestanden

Ergebnisse des Vorhabens

Evaluation und Zusammenfassung vorhandener Lösungen

Im Projekt ging es zunächst darum, den aktuellen Stand im Bereich der Provenance-Modellierung zu erfassen.

Als Ergebnis lässt sich feststellen, dass das einzige Provenance-Modell, das bereits standardisiert ist oder sich in der Standardisierung befindet, das PROV ist. PROV [18] ist eine Familie an Spezifikationen im Bereich Provenance. Die verschiedenen Spezifikationen wurden im Wesentlichen in den Jahren 2011 bis 2013 durch die W3C Provenance Working Group erarbeitet. Sie liegen als „W3C Recommendations“ vor.

Die wesentlichen und wichtigsten W3C-Spezifikationen zu PROV sind die folgenden:

- **PROV-O**: Die *PROV Ontology*. Diese Ontologie beschreibt formal die bei PROV verwendeten Begriffe und der zwischen ihnen bestehenden Beziehungen.
- **PROV-DM**: Das *PROV Data Model*. Dieses Datenmodell erlaubt die Abbildung von anwendungs- und domänenspezifischen Provenance-Informationen im Rechner und deren Austausch untereinander.
- **PROV-N**: Die *PROV Notation*. Diese Notation dient der menschenlesbaren Darstellung von Provenance-Informationen.
- **PROV-CONSTRAINTS**: Die Definition von *PROV Constraints*. Mit Hilfe von Constraints kann man Beschränkungen, Validität und Gültigkeit von Provenance-Informationen abbilden. Sie bilden eine Untermenge des PROV-DM.
- **PROV-AQ**: Die *PROV Provenance Access and Query* Spezifikation,

welche beschreibt wie man über Standard-Web-Protokolle auf Provenance-Informationen über Resourcen im Web zugreifen und diese finden kann.

- **PROV-PRIMER**: Eine einfache Einführung in das PROV Datenmodell.

Grundkonzpte von PROV

Die Grundkonzepte werden nun kurz vorgestellt. Die detaillierte Darstellung findet sich in ausgezeichneter Form in den Originalquellen zu PROV-DM und PROV-O sowie in relativ kurzer Form in dem PROV-PRIMER, einem Überblick über das PROV Datenmodell.

Die zugrundeliegende Definition von Provenance ist

> ***Provenance is information about entities, activities, and people involved in producing a piece of data or thing, which can be used to form assessments about its quality, reliability or trustworthiness.***

Dem entsprechend werden in PROV die drei Grundelemente *Entity*, *Activity* und *Agent* definiert. Zusätzlich werden Definitionen für Abhängigkeiten, Zusammenhänge, Rollen, Versionen und Revisionen, Plänen bzw. Rezepten und Zeit für die Grundelemente definiert. Abbildung 1 stellt die drei Grundelemente und einige grundlegende Beziehungen zwischen ihnen im Zusammenhang dar. Da PROV dazu dient, die Entstehung oder Verarbeitung von Dingen formal zu beschreiben, sind die Beziehungen in der Vergangenheitsform formuliert (z.B. für *wasDerivedFrom*: „Datei 2 wurde abgeleitet aus Datei 1“).

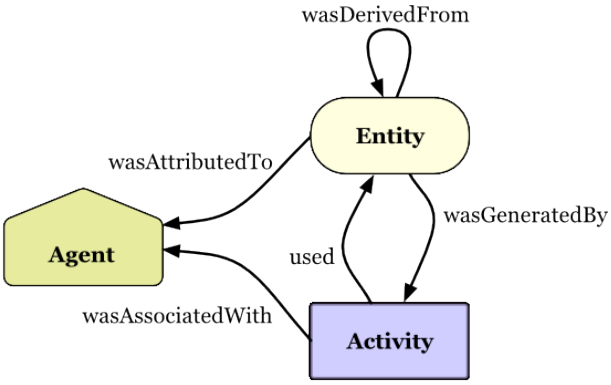


Abbildung 1: Grundkonzepte von PROV.

Die Grundelemente von PROV sind:

- **Entity**: Die *Entities* sind Dinge irgendeiner Form, zum Beispiel physikalische Objekte, digitale Objekte, Konzepte, oder sonstiges. Konkrete Beispiele sind Dateien, Dokumente, Webseiten, Grafiken etc.
- **Activity**: Die *Activities* erzeugen oder verarbeiten Entities. Activities können einzelne Aktionen, Operationen oder Prozesse sein.
- **Agent**: Die *Agents* haben bestimmte Rollen in Bezug auf die Activities; sie sind im gewissen Sinne für Activities verantwortlich. Agents können zum Beispiel Menschen, eine Software oder Organisationen sein.

Relationen setzen die Grundelemente zueinander in Beziehung. Die wichtigsten Relationen sind:

- **Benutzung und Erzeugung**: Die Relationen *used* und wasGeneratedBy beschreiben, was passiert ist. Dies sind die Benutzung einer Entity durch eine Acitvity und die Erzeugung einer Entity durch eine Activity.
- **Verantwortung**: Die Relationen *wasAttributedTo* und *wasAssociatedWith* beschreiben die Beziehung zwischen Agents und den

Entities bzw. Activities.

- **Ableitung und Revisionen**: Die Relation *wasDerivedFrom* beschreibt die Ableitung einer neuen Entity durch eine bestehende Entity. Die Relation *wasRevisionOf* beschreibt die Erzeugung einer neuen Version einer Entity.

Evaluation

Mit Hilfe dieser Grundelemente und den Relationen zwischen Ihnen kann man die Provenance verschiedenster Prozesse beschreiben. Bei der Evaluation im Rahmen dieses Vorhabens waren alle auftretenden Prozesse über das PROV Datenmodell abbildbar. Anhang von zwei Beispielen soll das im Folgenden illustriert und verdeutlicht werden:

- Softwareentwicklungsprozesse
- Gesundheitsrelevante Daten von Wearable Devices

Softwareentwicklungsprozesse

Softwareentwicklung ist heutzutage ein komplexer Prozess. Es sind eine Vielzahl von Entwicklungstools, Entwicklern und Prozessschritten auf verschiedenen Zeitskalen beteiligt (siehe zum Beispiel Abbildung 2). Das Ergebnis, die lauffähige Software, ist in Bezug auf ihre Qualität und Fehleranfälligkeit stark abhängig von dem Gesamt-Softwareentwicklungsprozess.

Die große Herausforderung bei Softwareentwicklungsprozess ist, den Prozess genau zu verstehen und möglichst Aussagen und Zusicherungen über die Qualität des Endprodukts, d.h. der lauffähigen Software, zu machen. Weitere Fragestellungen kommen aus dem Bereich *Complience*, also ob die Entwickler oder die Organisation sich an vorgegebene Prozesse oder Regularien gehalten

haben, oder aus dem Bereich Produktivität, wo es um die Analyse von Aufwänden geht. Tabelle 1 enthält Beispiele für Fragestellungen mit jeweils einer konkreten Frage, die sich mit Hilfe aufgezeichneter Provenance-Informationen beantworten lassen.

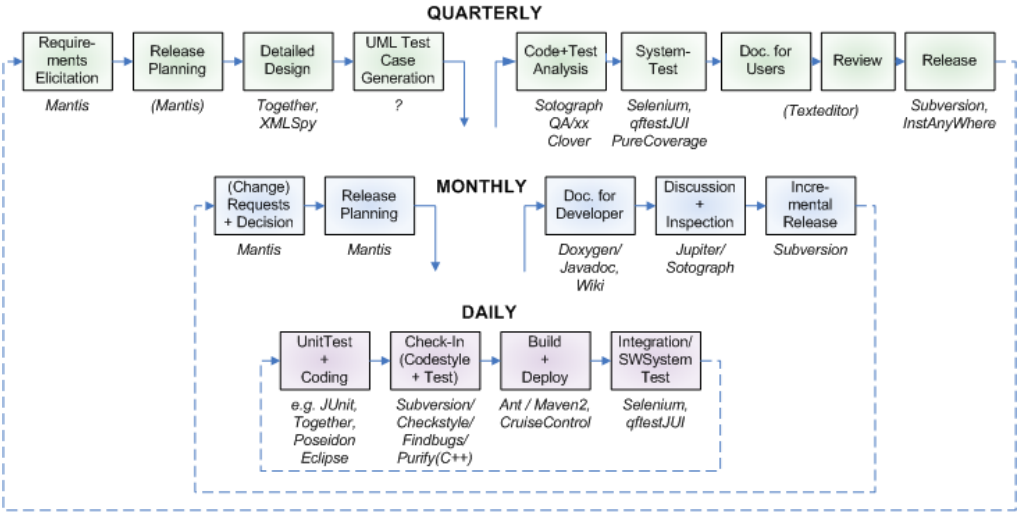


Abbildung 2: Typischer DLR-Softwareentwicklungsprozess

Fragestellung	Beispiel
Error detection	*Which change set resulted in more failing unit tests?*
Quality assurance	*How many releases have been produced this year?*
Process validation	*From which revision was release X built?*
Monitoring	*How much time has been spent implementing issue X?*
Statistical analysis	*How many developers contributed to issue X?*
Developer rating	*Which developer is most active in contributing documentation?*
Information	*Which features are part of release X?*

Tabelle 1: Fragestellungen für Softwareentwicklungsprozesse

Um die aufgezeigten Fragen bei Softwareentwicklungsprozessen mit Hilfe der Provenance beantworten zu können, muss der Prozess in dem PROV Datenmodell modelliert werden. Konkret wurde im Rahmen des Vorhabens ein Prozess mit bestimmten Entwicklungstools im DLR modelliert (siehe Abbildung 3), was sich aber leicht auf andere Softwareentwicklungsprozesse übertragen lässt.

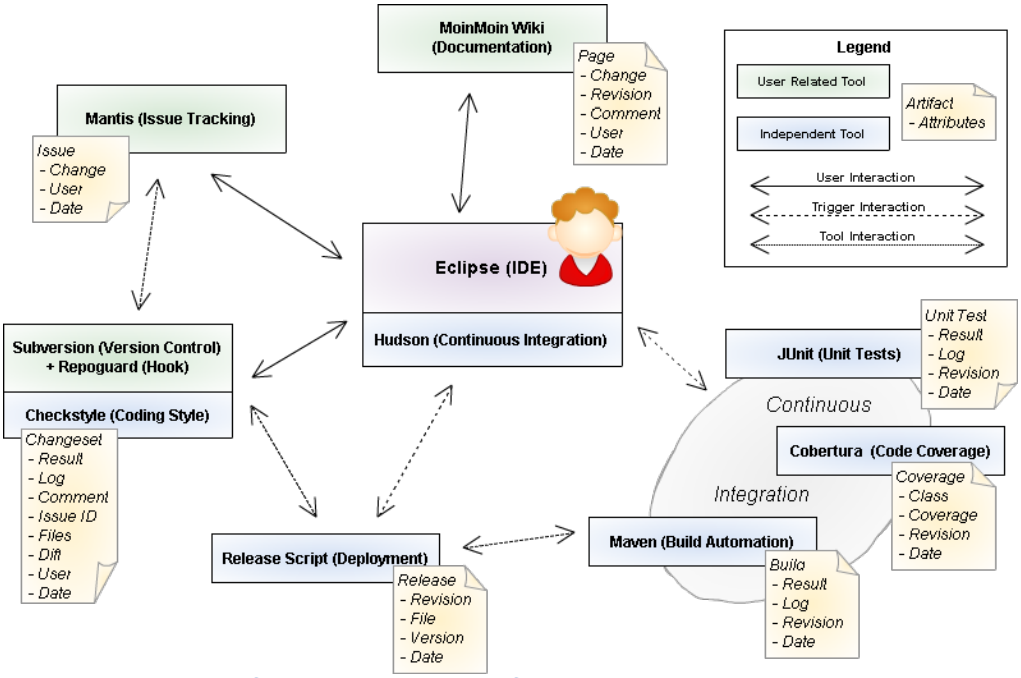


Abbildung 3: Beispiel für einen konkreten Softwareentwicklungsprozess aus dem DLR

Im Folgenden sind zwei Teilmodelle des PROV-Datenmodells dargestellt , wobei es hier nicht um Vollständigkeit geht sondern nur um zwei Beispiele. Abbildung 4 zeigt das Teilmodell für die Änderung eines *Issues*, also zum Beispiel einer Anforderung oder eines Fehlerreports. Abbildung 5 zeigt das Teilmodell für den *Commit* einer Source-Code-Änderung, d.h. das Übergeben einer Dateiänderung an das zentrale Versionsmanagementsystem.

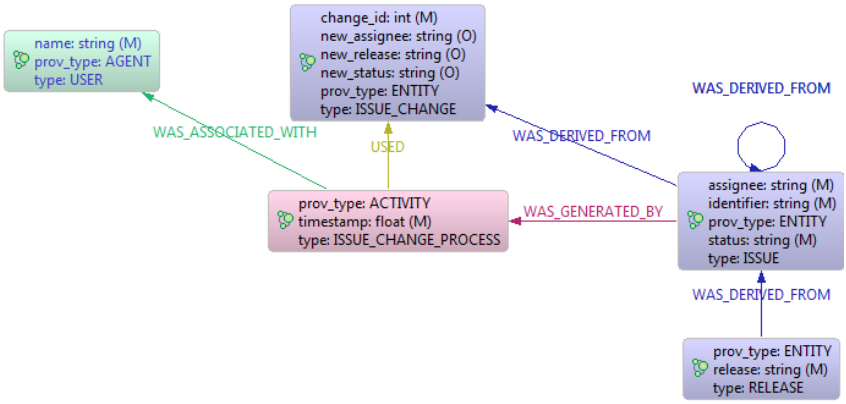


Abbildung 4: PROV Modell für die Änderungen eines Issues (z.B. Anforderung oder Bug)

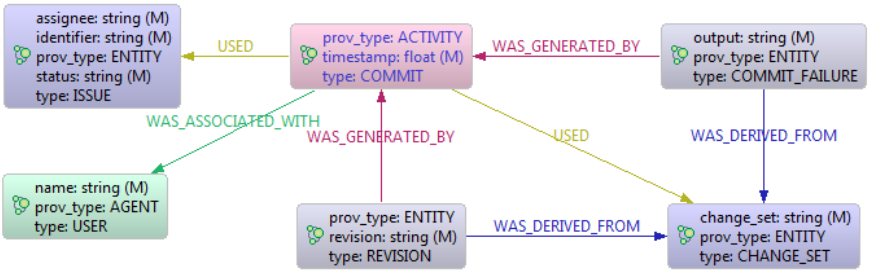


Abbildung 5: PROV Modell für einen Commit, d.h. einer Änderung an einem Source Code

Diese beiden Beispiele sollen zeigen, wie die konkrete Modellierung für (Teil)-Prozesse mit PROV-DM aussieht. Zur Nutzung in der Praxis muss man als nächstes die im PROV Modell geforderten Informationen zur Laufzeit des Softwareentwicklungsprozesses besorgen und in einem *Provenance-Store* speichern. Danach kann man auf den gespeicherten Informationen abfragen machen.

Gesundheitsrelevante Daten von *Wearable Devices*

Das zweite Evaluationsbeispiel ist aus dem Bereich gesundheitsrelevanter, persönlicher Daten die mit Hilfe von sog. *Wearables* erfasst und anschließend in der Cloud gespeichert und auf verschiedene Weise analysiert werden. Das Grundproblem, das hier durch Provenance gelöst werden soll, ist die Nachvollziehbarkeit dieser Daten. Also welchen Weg haben die Daten genommen, wo wurden Sie gespeichert, von wem wurden sie verarbeitet, welchen Organisationen hatten Zugriff auf die Daten, etc. In Tabelle 2 sind Beispiele für konkrete Provenance-Fragen aufgelistet, die wir in unserer Anforderungsanalyse ermittelt haben [19].

Fragestellung	Beispiele
Entity focused	• *What data about the user were created during the activity X?* • *What data about the user were automatically generated?* • *What data about the user were derived from manual input?*
Activity focused	• *Which activities support visualization of the users data?* • *In which activities can the user input data?* • *What processes are communicating data?*
Agent focused	• *What parties were involved in generating data X?* • *What parties got access on data X?* • *Can other parties see user's data X?*

Tabelle 2: Fragestellungen für gesundheitsrelavante Daten von Wearable Devices

Die Abbildung 6 zeigt ein Bespiel für solche Daten. Dargestellt ist die Erfassung von Gewicht mit einer elektronischen Waage („Withings Scale"), die Weiterleitung des gemessenen Wertes über WLAN zur der Cloud des Herstellers, die Darstellung des Wertes in einer App auf dem Smartphones des Nutzern, die Auswertung der Gewichtswerte mit Apps und Web-Services von Drittanbietern oder selbst geschriebenen Skripten, bis hin zur Veröffentlichung von

Gewichtswerten auf Social-Media-Kanälen. Ähnliche Beispiel für Daten-Workflows lassen sich für viele andere Messwerte aufstellen, die mit modernen elektronisch vernetzen Sensoren erfasst werden können.

Da diese Workflows sehr komplex und vielfältig werden können, waren sie ein gutes Beispiel für die Evaluation des PROV-DM.

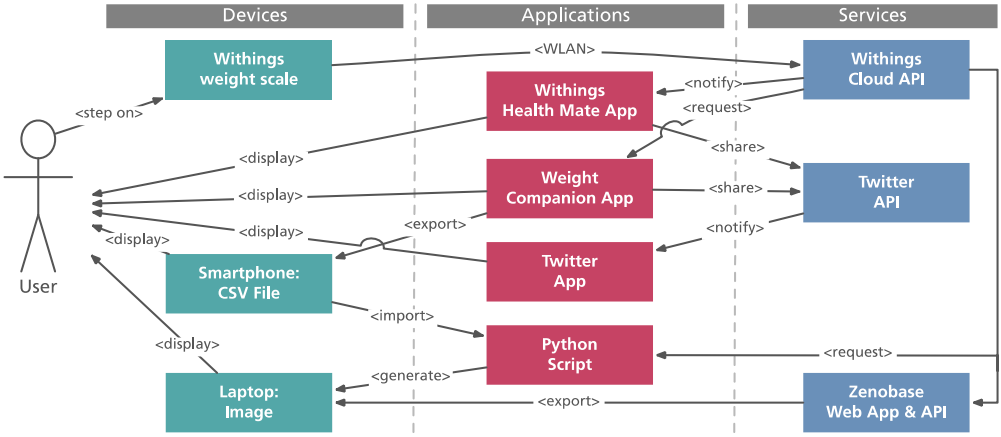


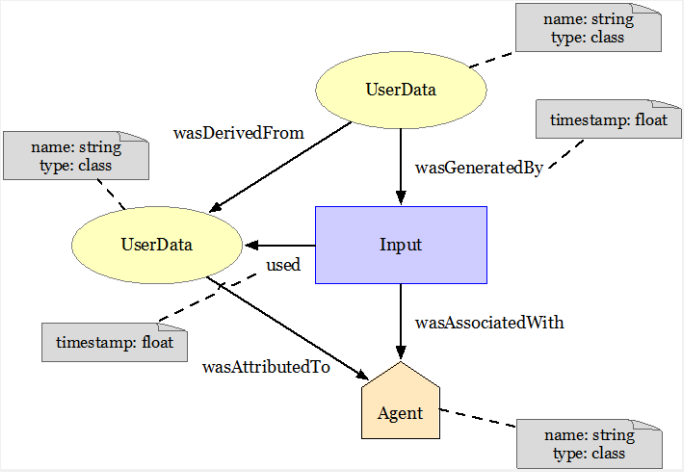
Abbildung 6: Datenfluss für die Erfassung und Verarbeitung von Gewichtsdaten

Als Ergebnis der Evaluation ist eine erste Version eines allgemeines Provenance-Modell in PROV-DM herausgekommen, mit dem sich viele der relevanten Workflows abbilden lassen. Tabelle 3 zeigt das entstandene Modell [19, 20], aufgeteilt in Submodelle für die fünf wesentlichen Aktivitäten:

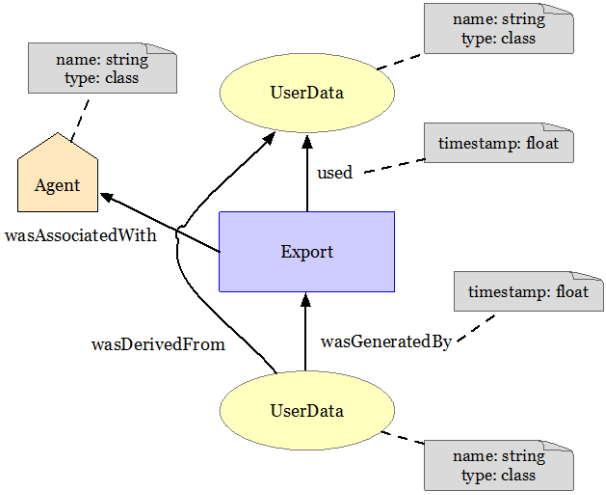
- Input
- Export
- Request
- Aggregate
- Visualize

Aktivität PROV Submodel

Input

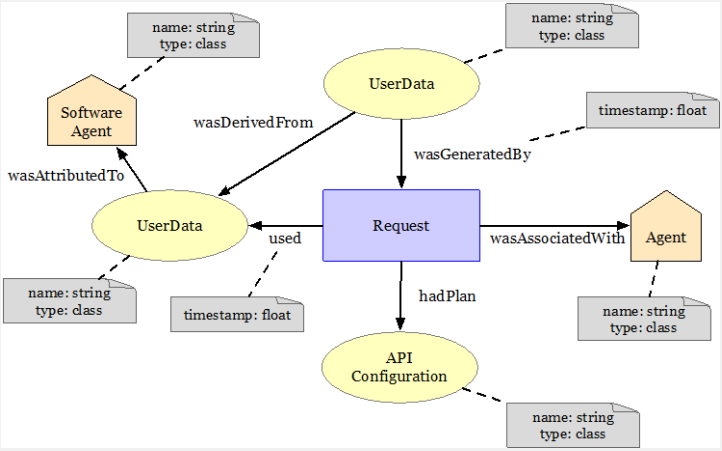


Export

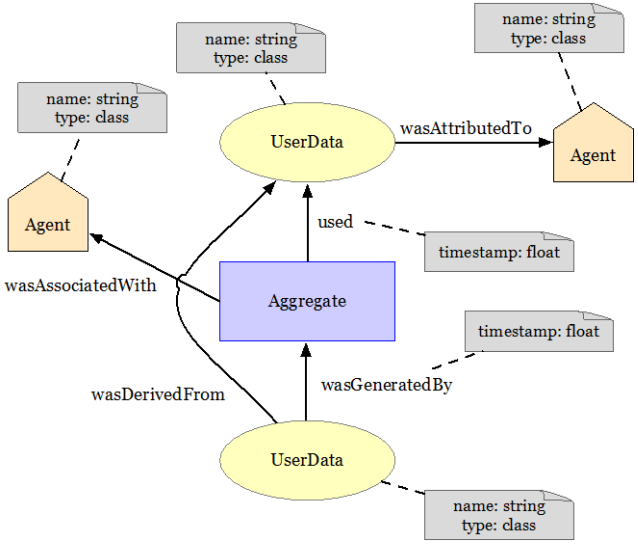


Aktivität PROV Submodel

Request



Aggregate



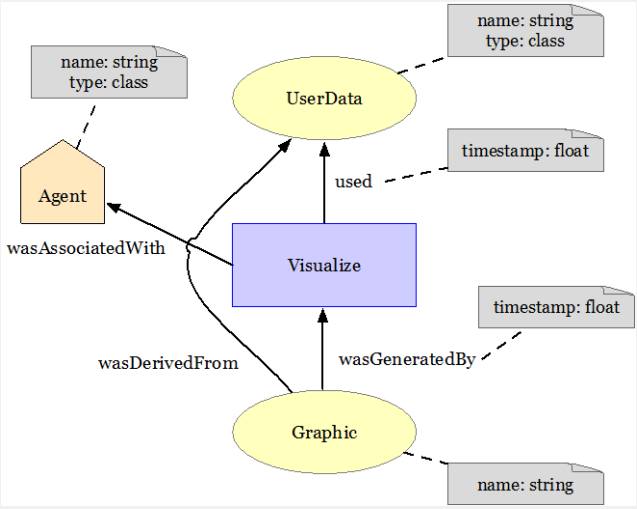
Aktivität	PROV Submodel
Visualize	

Tabelle 3: PROV Datenmodell für Daten von Wearables. Zur besseren Übersicht aufgeteilt in Submodelle für jede Aktivität.

Zum Demonstrieren, wie die aufgezeichnete Provenance-Information in der Praxis aussieht, hier ein Beispiel: Abbildung 7 zeigt die Provenance für die Erzeugung eine Bildes aus Schrittzählerdaten. Diese Provenance wurde automatisch aufgezeichnet. Das Programm holt sich Schrittzählerdaten von der Cloud des Herstellers („library:fitbit“) und verarbeitet diese in mehreren Schritten unter Benutzung verschiedener Bibliotheken bis am Ende eine Grafik als Ergebnis herauskommt („graphic:steps“).

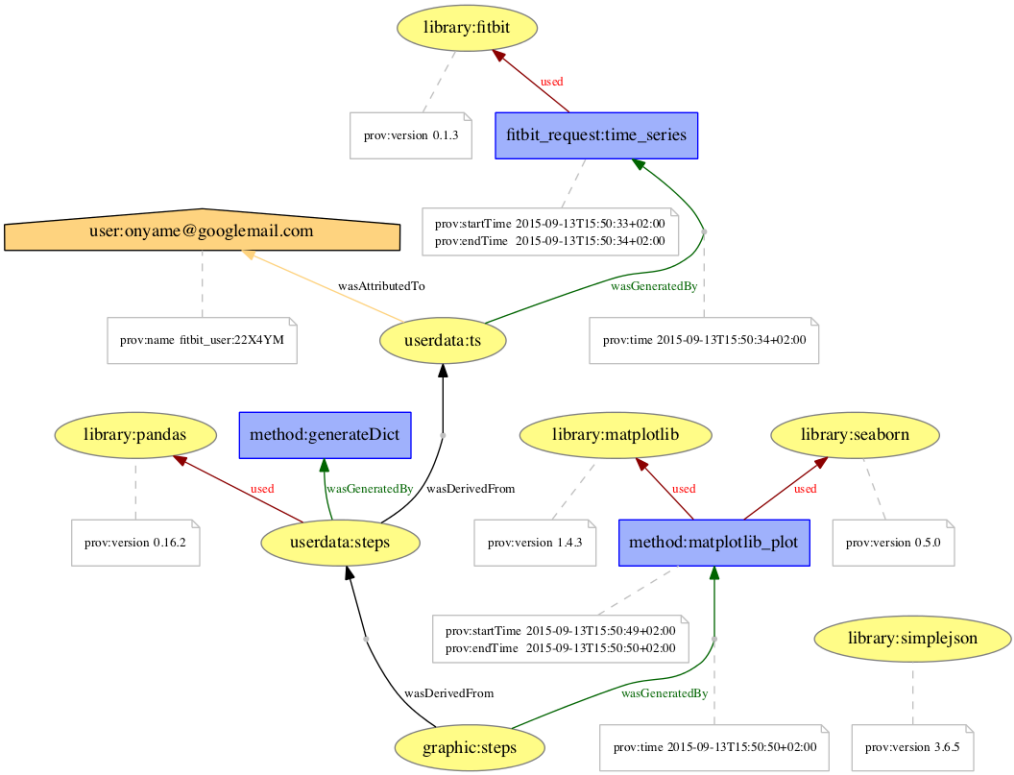


Abbildung 7: Aufgezeichnete Provenance der Analyse von Schhrittzählerdaten

Weiterentwicklung und Bereitstellung der Referenzimplementierung

Um Provenance-Informationen zu nutzen, müssen Sie während der Laufzeit von Prozessen gespeichert werden. Dies kann auf vielfältige Weise geschehen, zum Beispiel in klassischen Dateien oder in beliebigen Datenbanken.

Damit Nutzer eine leicht zugängliche Speichermöglichkeit für Provenance-Informationen zur Verfügung haben, wurde im Rahmen des Vorhabens eine

Provenance-Datenbank als Referenzimplementierung erstellt. Diese Referenzimplementierung PyProv steht unter einer Open-Source-Lizenz (Apache Licence 2.0) allen auf der Plattform github zur Verfügung .

Die Implementierung PyProv stellt eine REST-Schnittstelle zur Verfügung, über die Provenance-Informationen aus Auswendungen und Workflows heraus geschickt werden können. Die Provenance-Informationen werden dann in eine Graph-Datenbank Neo4j gespeichert. Eine Graph-Datenbank wird deshalb verwendet, weil Provenance-Informationen Graphen sind (gerichtete azyklische Graphen).

Die University of Southampton hat einen öffentlich zugänglichen Provenance-Store aufgesetzt und zur Verfügung gestellt [21]. Dieser ist über eine Web-Service-Schnittstelle benutzbar. In diesem Provenance-Store können Provenance-Dokumente in der PROV-N Notation gespeichert werden. Die Dokumente können visualisiert und in verschwenden Formaten exportiert werden. Für einen einfachen Test und für Experimente mit Provenance ist dies eine sehr komfortable Möglichkeit. Lediglich für vertrauliche oder geheime Daten sollte bzw. muss ein eigener Provenance-Store innerhalb der eigenen Organisation aufgesetzt werden.

Ausblick

Die Nutzung und die Entwicklung weiterer grundlegender Technologien wird in einer Reihe von Arbeiten und Projekten weitergeführt. Dabei wird bei allen aktuellen Arbeiten das PROV Datenmodell verwendet. Konkret wird an folgenden Themen im DLR gearbeitet:

- **Integration in reale Anwendungen**: Hier wird das Aufzeichnen von Provenance-Informationen in Anwendungen und Softwaresysteme integriert. Dabei wird dann die Provenance ständig zur Laufzeit

aufgezeichnet. Beispiele sind die Integration in Apps und Services aus dem Gesundheits- und Medizinbereich auf Basis des Provenance-Modells für Quantified Self [22] und die Integration in das System BACARDI zur Prozessierung und Speicherung von Weltraumschrottdaten.

- **Integration in Messaging-Systeme**: Als Grundlage zur Aufzeichnung von Provenance in Anwendungen aus dem Bereich der verteilten Systeme und des Internet-of-Things werden die beiden Messaging-Systeme *MQTT* und *ZeroMQ* um Provenance-Aufzeichnung erweitert.
- **Visualisierung von Provenance**: Um Provenance-Informationen für Menschen nutzbar zu machen, kann sie visualisiert werden. Hier werden grundlegenden Forschungen durchgeführt, um diese Graph-basierten Daten geeignet zu visualisieren. Als eine Beispiel werden die Provenance-Informationen aus dem Medizinbereich in Form von *Comics* visualisiert. Damit sollen die Informationen Leuten zugänglich gemacht werden, welche von IT keinerlei Kenntnisse haben.
- **Speicherung von Provenance**: Zur Speicherung von Provenance wird vor allem dran gearbeitet, leicht benutzbare Schnittstellen in Python zu Graph-Datenbanken zu entwickeln. Neben Neo4j sollen hierbei auch weitere Graph-Datenbanken wie zum Beispiel ArangoDB genutzt werden können. Zur fälschungssicheren Speicherung von Provenance werden *Blockchains* verwendet. Mit diesen verteilten, kryptographisch gesicherten Datenbanken sollen insbesondere Provenance-Informationen von sensiblen Anwendungen gesichert werden.

Abbilungsverzeichnis

Tabellenverzeichnis

Literaturverzeichnis

[1] L. Moreau, P. Groth, S. Miles, J. Vazquez-Salceda, J. Ibbotson, S. Jiang, S. Munroe, O. Rana, A. Schreiber, V. Tan, and L. Varga, "The provenance of electronic data," *Communications of the Acm,* vol. 51, no. 4, pp. 52-58, Apr, 2008.

[2] L. Moreau, B. Clifford, J. Freire, J. Futrelle, Y. Gil, P. Groth, N. Kwasnikowska, S. Miles, P. Missier, J. Myers, B. Plale, Y. Simmhan, E. Stephan, and J. Van den Bussche, "The Open Provenance Model core specification (v1.1)," *Future Generation Computer Systems-the International Journal of Grid Computing and Escience,* vol. 27, no. 6, pp. 743-756, Jun, 2011.

[3] K. Belhajjame, R. B'Far, J. Cheny, S. Coppens, S. Cresswell, Y. Gil, P. Groth, G. Klyne, T. Lebo, J. McCusker, S. Miles, J. Myers, S. Sahoo, and C. Tilmes, *PROV-DM: The PROV Data Model,* Technical Report, World Wide Web Consortium, 2013.

[4] M. Ney, *Enabling a data management system to support the "good laboratory practice",* Masterarbeit, Freie Universität Berlin, 2011.

[5] M. Ney, G. K. Kloß, and A. Schreiber, "Using Provenance to Support Good Laboratory Practice in Grid Environments," *Data Provenance and Data Management in eScience*, pp. 157-180, Berlin Heidelberg: Springer, 2013.

[6] L. Moreau, B. Ludascher, I. Altintas, R. S. Barga, S. Bowers, S. Callahan, G. Chin, B. Clifford, S. Cohen, S. Cohen-Boulakia, S. Davidson, E. Deelman, L. Digiampietri, I. Foster, J. Freire, J. Frew, J. Futrelle, T. Gibson, Y. Gil, C. Goble, J. Golbeck, P. Groth, D. A. Holland, S. Jiang, J. Kim, D. Koop, A. Krenek, T. McPhillips, G. Mehta, S. Miles, D. Metzger, S. Munroe, J. Myers, B. Plale, N. Podhorszki, V. Ratnakar, E. Santos, C. Scheidegger, K. Schuchardt, M. Seltzer, Y. L. Simmhan, C. Silva, P. Slaughter, E. Stephan, R. Stevens, D. Turi, H. Vo, M. Wilde, J. Zhao, and Y. Zhao, "Special issue: The first provenance challenge," *Concurrency and Computation-Practice & Experience,* vol. 20, no. 5, pp. 409-418, Apr 10, 2008.

[7] A. Bachmann, H. Bergmeyer, and A. Schreiber, "Evaluation of aspect-oriented frameworks in Python for extending a project with provenance

documentation features," *The Python Papers,* vol. 6, no. 3, 2011.
[8] M. Kunde, H. Bergmeyer, and A. Schreiber, "Requirements for a Provenance Visualization Component," *Provenance and Annotation of Data and Processes. IPAW 2008. Lecture Notes in Computer Science.* J. Freire, D. Koop and L. Moreau, eds., pp. 241-252: Springer-Verlag, 2008.
[9] M. Kunde, "Visualization-Panel for Provenance Data," Bachelorarbeit, Informatik, Fontys Hogeschool Techniek en Bedrijfsmanagement, 2008.
[10] G. K. Kloß, and A. Schreiber, "Provenance Implementation in a Scientific Simulation Environment," *Provenance and Annotation of Data. IPAW 2006. Lecture Notes in Computer Science,* vol. 4145, pp. 37-45, 2006.
[11] M. Ney, "Enabling a Data Management System to Support the "Good Laboratory Practice"," Masterarbeit, Freie Universität Berlin, 2011.
[12] M. Ney, G. K. Kloss, and A. Schreiber, "Using Provenance to Support Good Laboratory Practice in Grid Environments," *Data Provenance and Data Management in eScience*, 426, Q. Liu, Q. Bai, S. Giugni, D. Williamson and J. Taylor, eds., pp. 157-180, Berlin, Heidelberg: Springer Berlin Heidelberg, 2013.
[13] C. Teichmann, "Analysis of Software-Engineering-Processes," Masterarbeit, Computer Science, Technische Universität Ilmenau, Berlin, 2013.
[14] H. Wendel, "Using Provenance to Trace Software Development Processes," Masterarbeit, Institut für Informatik III, Universität Bonn, 2010.
[15] H. Wendel, M. Kunde, and A. Schreiber, "Provenance of Software Development Processes," *Provenance and Annotation of Data and Processes: Third International Provenance and Annotation Workshop, IPAW 2010, Troy, NY, USA, June 15-16, 2010. Revised Selected Papers*, 6378, D. L. McGuinness, J. R. Michaelis and L. Moreau, eds., pp. 59-63, Berlin, Heidelberg: Springer Berlin Heidelberg, 2010.
[16] C. Bochner, R. Gude, and A. Schreiber, "A Python Library for Provenance Recording and Querying," *Provenance and Annotation of Data and Processes. IPAW 2008*. Lecture Notes in Computer Science, vol. 5272, pp. 229-240, 2008.
[17] A. Schreiber, M. Ney, and H. Wendel, "The Provenance Store prOOst for the Open Provenance Model," *Provenance and Annotation of Data and Processes: 4th International Provenance and Annotation Workshop, IPAW 2012, Santa Barbara, CA, USA, June 19-21, 2012, Revised Selected Papers*, Berlin, Heidelberg, 2012, pp. 240-242.

[18] L. Moreau, and P. Groth, *Provenance: An Introduction to PROV*, pp. 129: Morgan & Claypool, 2013.
[19] B. Janisch, "Developing an abstract Quantified Self Provenance-model," Master project, Angewandte Informatik, Hochschule Bonn-Rhein-Sieg, Sankt Augustin, 2015.
[20] A. Schreiber, "A Provenance Model for Quantified Self Data," *Universal Access in Human-Computer Interaction. Methods, Techniques, and Best Practices, UAHCI 2016, Part I, LNCS,* vol. 9737, 2016.
[21] T. D. Huynh, and L. Moreau, "ProvStore: A Public Provenance Repository," *Provenance and Annotation of Data and Processes: 5th International Provenance and Annotation Workshop, IPAW 2014, Cologne, Germany, June 9-13, 2014. Revised Selected Papers*, pp. 275-277, 2015.
[22] A. Schreiber, and D. Seider, "Towards Provenance Capturing of Quantified Self Data," *Provenance and Annotation of Data and Processes, IPAW 2016, LNCS*, vol. 9672, 2016.

0939-2963

ISRN DLR-FB—2016-4